UMIACS-TR-90-54 April 1990
CS-TR -2454

Place/Transition Nets with Debit Arcs

P. David Stotts*
Institute for Advanced Computer Studies and
Department of Computer Science
University of Maryland
College Park, MD 20742

Parke Godfrey†
Department of Computer Science
University of Maryland
College Park, MD 20742

Abstract

We add an extension called *debit arcs* to traditional place/transition nets. A debit arc allows its destination transition to fire whenever desired, but records a *debt* (or *antitoken*) in its source place if a token there is not consumed. A normal token can annihilate with an antitoken, which can be thought of as "paying off" the debt. Two natural rules for token/antitoken annihilation (*instantaneous*, and *delayed*) are examined and are shown to create two distinct classes of automaton in terms of language recognition power. Under instantaneous annihilation, nets with debit arcs are equivalent as a class to Turing machines, and so extend the modeling power of standard place/transition nets. Under delayed annihilation, nets with debit arcs are equivalent as a class to standard place/transition nets, and thus are only a notational convenience. Nets with debit arcs are shown to be a special case of colored nets.

Key words: Petri nets, place/transition nets, automata theory, formal languages, parallel computation model, colored nets, high-level nets. (KR) (—

CR categories: F.1.1 (Petri nets)

*Supported in part by the Center for Excellence in Space Data and Information Sciences at the NASA Goddard Space Flight Center.

†This research has been supported in part by the U.S. Army Institute for Management Information and Computer Science (AIRMICS).

1 Basic definitions

We start with a basic place/transition net as described in Reisig [4], with unit weights and infinite capacities on all arcs. Within this environment, we define an extended form of arc called a *debit arc*, as follows:

Definition 1 Debit arc
A *debit arc* is a net arc (s, t) that can alter the normal execution behavior of a net. We define the *debitset* of t as $\bullet t \supseteq \rhd t = \{s \mid (s,t) \text{ is a debit arc in the net }\}$. At any time, t is enabled to fire if $\bullet t - \rhd t$ is marked. Firing t adds a token to each place in $t\bullet$, and removes a token from each place in $\bullet t - \rhd t$. In addition, for each $s \in \rhd t$, if s is not marked then firing t creates in s a *debt*, also called an *antitoken*. If s is marked, then firing t may either consume a token or create a debt. The choice of whether to create debt or not may be constrained (as described below).

In this paper we will refer to a net containing one or more debit arcs as a *debit net*. To represent debts we extend the definition of *marking* to contain both token and antitoken counts for places:

Definition 2 Marking
A *marking* of a debit net is the normal total function $M : S \rightarrow integer$ to give the token count for each place, along with another total function $D : S \rightarrow integer$ that gives the antitoken count for each place that is the source of a debit arc.[1] A *state* is then the vector of integer pairs $(M(s_i), D(s_i))$ covering all places s_i in the net.

An example of debit arc behavior is shown in Figure 1. The arc between place s_1 and transition t_1 is a debit arc, drawn as a line with an open triangle for an arrowhead. Firing t_1 creates an antitoken in s_1, drawn as an open circle. The figure shows a state change from $[(0,0)(1,0)(0,0)]$ to $[(0,1)(0,0)(1,0)]$. Note that since debit arcs act like normal arcs in the presence of tokens, the state change from $[(1,0)(1,0)(0,0)]$ to $[(0,0)(0,0)(1,0)]$ is valid for the net structure shown. We now define how antitokens can be eliminated from a debit net.

Definition 3 Annihilation
A token and an antitoken residing in the same place can *annihilate* each other. Annihilation is possible in any state in which $M(s) > 0$ and $D(s) > 0$ for some place s. The annihilation causes a state change in which the pair $(M(s), D(s))$ becomes $(M(s) - 1, D(s) - 1)$.

In the remainder of this paper, we will discuss two simple annihilation policies, each leading to different recognition power for debit nets. The first policy is termed *instantaneous annihilation*, and specifies that whenever a token and antitoken become co-resident in a place, they must immediately annihilate. The second policy is termed *delayed annihilation*, and specifies that an annihilation need not occur in the first state it is possible, but may take place in any subsequent possible state instead. Note that the delayed annihilation policy allows a token to come into a place containing an antitoken, and later leave that place without an annihilation occurring at all. As an example of

[1]By convention we define $D(s)$ to be a constant 0 for any s that is not the source of a debit arc. Having D be a total function is a convenience, and saves a change of notation in later extensions to these early definitions.

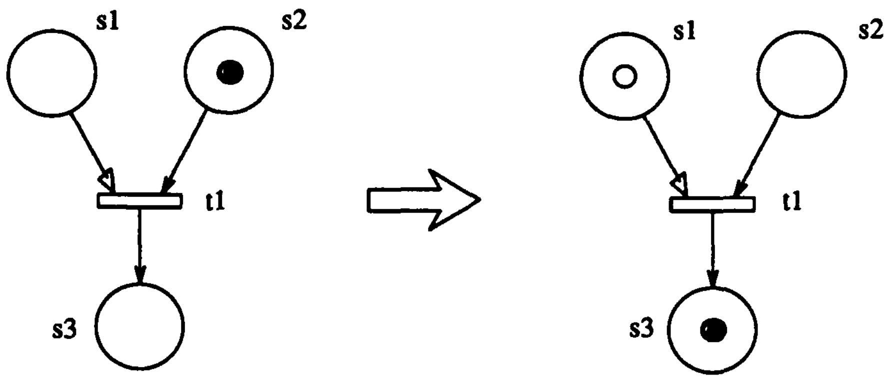


Figure 1: Firing a transition with a debit arc.

how annihilation policies can constrain execution state sequences, refer again to the net structure shown in Figure 1, and consider the marking $[(1,1)(1,0)(0,0)]$ for it. Under delayed annihilation, either of states $[(0,1)(0,0)(1,0)]$ or $[(0,0)(1,0)(0,0)]$ is a valid next state. Under instantaneous annihilation, only state $[(0,0)(1,0)(0,0)]$ is a valid next state.

2 Recognition power of debit nets

Several methods are common for ascribing languages to nets, differing in how symbols are associated with transitions and how final states are defined. For this discussion, we assume that whenever a transition is fired, whether it thereby creates a debt or not, the symbol mapped to the transition is added to the string being generated.[2] In this case, the three methods described by Peterson [3] for mapping symbols to transitions are still applicable. However the definition of final states can be extended because of our new view of state. Now one must define a final state in terms of both tokens and antitokens in places. Perhaps the simplest (and most obvious) way to do this is to require all debts to be paid in any final state, that is, no antitokens may exist in a final state. This interpretation views a debt as a future enabling of a past event, and it requires that the enabling must actually come about for the generated string to represent a valid computation of the net (since the generated string contains a transition symbol for the event).

2.1 Under delayed annihilation

Under the delayed annihilation policy, debit nets are a notational convenience only, and do not extend the modeling power of place/transition nets.

Theorem 1:
The class of debit nets under the delayed annihilation policy is equivalent to the class of normal place/transition nets.

Proof: by double subset.

[2]See section 4 for a discussion of other ways to have debts affect the strings generated by a net.

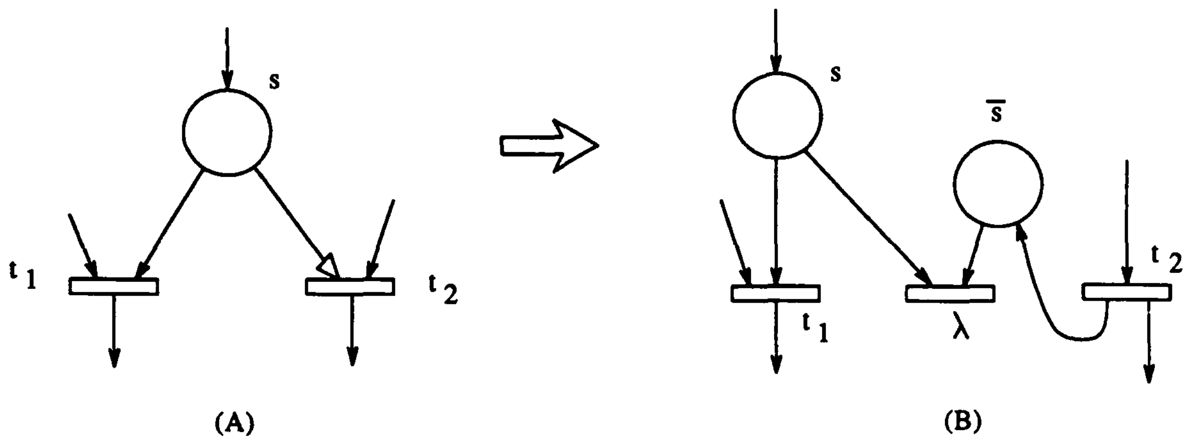


Figure 2: Debit net represented as normal place/transition net.

First, it is trivially the case that all normal place/transition nets are debit nets (with no debit arcs).

Secondly, we show the other direction with a translation to convert any debit net under delayed annihilation into a normal place/transition net that generates the same language. An example of this translation is shown in Figure 2. In net B the firing of t_2 is not constrained by s (but is still constrained by any other of its input places). Firing t_2 records a "debt" in a new place $\bar{s}$. A new transition λ is included to allow cancellation of "debt" tokens in $\bar{s}$ with "regular" tokens in s. In B we map a null symbol to λ, and map the same symbols to t_1 and t_2 as in net A. Then the language generated by net A is the same as the language generated by net B, as follows. In net A three different events are possible: firing t_1; firing t_2 to make a debt; and annnihilation. The event of firing t_2 to consume a token in s is no different from firing t_2 to make a debt and then immediately annihilating the antitoken with the token in s. We now consider the effect of each possible event on the generated string.

case 1: firing t_1
In net A, t_1 can fire whenever $M(s) > 0$ and all of its other input places are marked; in net B, the preconditions of this event are identical. Thus firing t_1^{A} is simulated by firing t_1^{B}, and *vice versa*. In both cases the same symbol is added to the generated string.

case 2: firing t_2 to make a debt
In net A, t_2 can fire whenever $\bullet t_2 - \{s\}$ is marked; in net B the same conditions hold (trivially, since $s \notin \bullet t_2$). Thus firing t_2^{A} to make a debt is simulated by firing t_2^{B}, and *vice versa*. In both cases the same symbol is added to the generated string.

case 3: annihilation of token and antitoken
In net A, in any state in which $M(s) > 0$ and $D(s) > 0$, an annihilation can occur, adding no symbols to the generated string. First, whenever $M(s) > 0$ in A it is the case that $M(s) > 0$ in B. Annihilation in A is then simulated in B by firing transition λ, which also adds no symbols to the

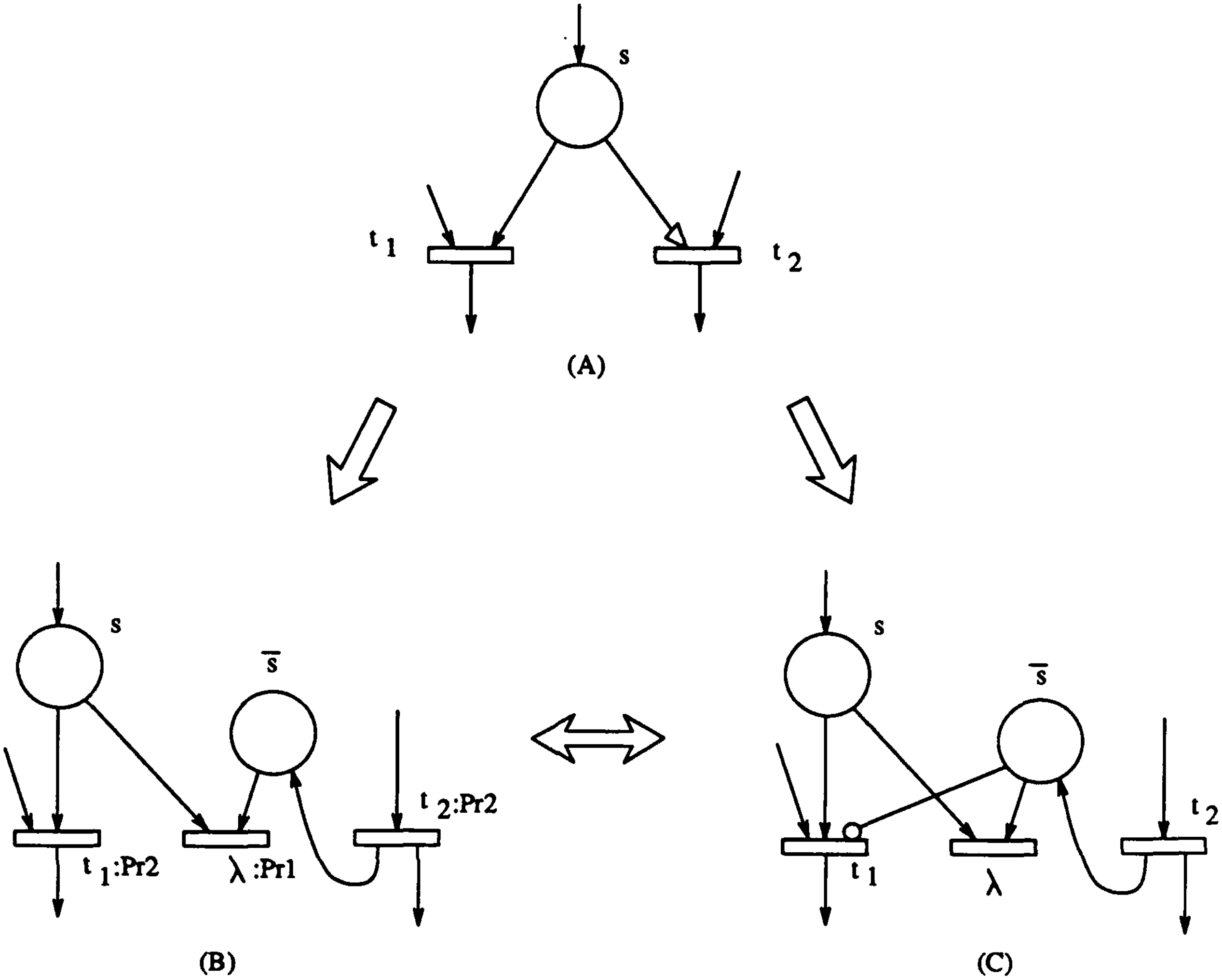


Figure 3: Modeling (A) debit arcs with (B) priorities or (C) inhibitor arcs.

generated string. In A, $D(s) > 0$ implies that transition t_2^A has previously fired; thus, in net B it must be the case that $M(\bar{s}) > 0$, and so transition λ will be enabled in B whenever an annihilation is possible in A. In the other direction, whenever λ fires in B (having no effect on the generated string) it is simulated by an annihilation in A.
□

2.2 Under instantaneous annihilation

We now consider debit nets under the instantaneous annihilation policy, which means that tokens and antitokens cannot coexist in the same place; whenever a token and antitoken meet, they annihilate. This implies that any place s can only have markings of the form $(n, 0)$ or $(0, n)$ for $n \in \{0, 1, 2 \ldots\}$. We can formalize this annihilation rule by mandating that whenever a state change occurs from a transition firing, the marking (n, m) at place s is immediately adjusted to $(n \dot{-} m, m \dot{-} n)$, where

$$n \dot{-} m = \begin{cases} n - m, & \text{if } n - m \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

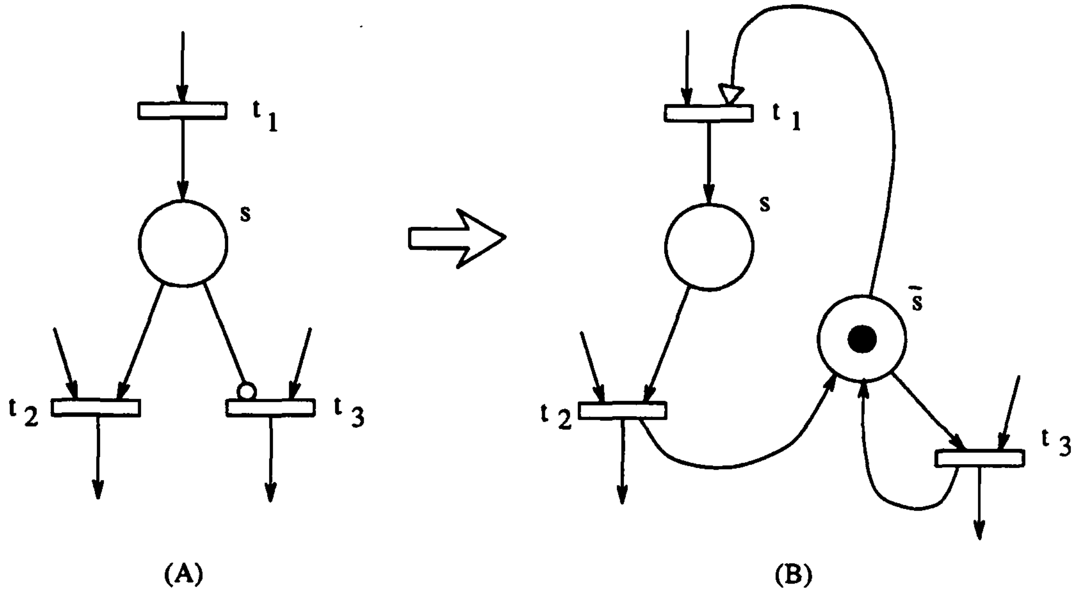


Figure 4: Inhibitor arc as debit net with instantaneous annihilation

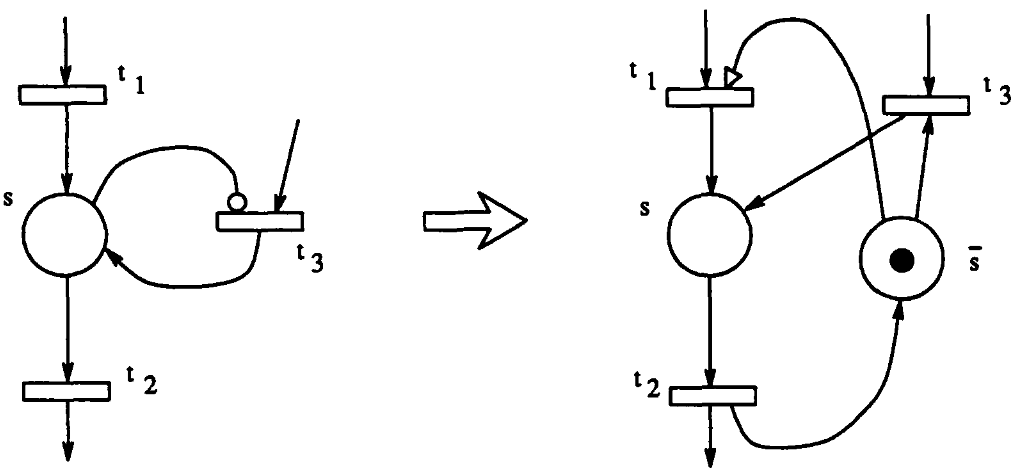
Since either $M(s)$ or $D(s)$ is always 0, we can dispense with the more general pair notation when employing the instantaneous annihilation rule. It is more convenient to represent the marking at each place by a single integer which can take on negative values. The value $n \geq 0$ represents n tokens at s, whereas $-n$ represents n antitokens at s. We can then refer to the state at place s simply as $M(s)$ where $M(s)$ can take on negative values. In the following discussion, let M_i denote the net state immediately before the firing of a given transition and M_f denote the net state immediately after. A debit arc now has very simple semantics. If $s \in \triangleright t$ and t is fired, the condition

$$M_f(s) = M_i(s) - 1$$

must hold (assuming $t \notin \bullet s$). Of course, if $s \in \bullet t - \triangleright t$ for some t, and $M(s) \leq 0$, then t is not enabled.

In debit nets with instantaneous annihilation, token counts can go negative at designated places. Debit arcs are thus a means by which an event can be allowed to proceed even if all its enabling conditions are not (yet) met; doing so records a debt that is canceled when the enabling condition next holds. At first it is not readily obvious that this increases a net's computational power. For instance, it is not trivial to show how a zero test can be performed in a debit net.

We have previously shown that the class of debit nets under the delayed annihilation rule is, indeed, no more powerful computationally than the class of normal place/transition nets. Thus far, debit arcs have only proven to be a notational convenience. The concept of allowing debts in places alone does nothing to increase the computational power of Petri nets either. For example, we could allow one to initially mark places with antitokens. This can easily be shown to provide no increase in recognition power. If we adopt an instantaneous annihilation policy, then the initial debts must be paid at a place before it can hold tokens and subsequently enable transitions. This too can easily be shown to be equivalent to the standard place/transition net class. Intuitively, this is because there exists no way to introduce debt dynamically. The only debt is that which exists in the initial marking.



(A) (B)

Figure 5: Degenerate case and its representation as debit net.

Once we allow a net to have debts created during execution, though, and we adopt an instantaneous annihilation policy, the situation changes. Figure 3 shows a debit subnet (A) expressed in terms of an equivalent subnet (B) employing priorities on transitions, and another equivalent subnet (C) employing inhibitor arcs.

In subnets B and C, net A's place s is represented by two places, s and $\bar{s}$. Place $\bar{s}$ can be thought of as harboring the "antitokens". Since we want tokens and antitokens to annihilate, we add a transition λ to act as a sink. To accomplish *immediate annihilation*, in B we give λ a higher priority than all other (non-*lambda*) transitions in the net. In C we have $\bar{s}$ inhibit all other transitions until its "antitokens" are all gone.[3]

Intuitively, then, it would seem that instantaneous annihilation does increase the modeling power of debit nets. The following theorem establishes this fact more formally.

Theorem 2:
The class of debit nets under the instantaneous annihilation policy is equivalent to the class of Turing machines (TM-equivalent).

Proof:
We show this by providing a universal transformation from Petri nets with inhibitor arcs into Petri nets with debit arcs. This transformation is represented in Figure 4. We prove the behavior of these two net constructs to be identical. Once this is established, we will have demonstrated a clear method for rewriting any Petri net using inhibitor arcs into one using debit arcs. Thus the class of debit nets represents a superset of the class of Petri nets with inhibitor arcs. The class of Petri nets with inhibitor arcs is also known to be TM-equivalent [1]. These facts together with Church's Thesis are sufficient to show that the class of Petri nets with debit arcs is TM-equivalent.

[3]Note that if we remove the priorities in net B and the inhibitor arc in C in Figure 3, then the nets have behavior equivalent to the debit net in A under the delayed annihilation policy. We no longer force tokens and "antitokens" to be canceled out before proceeding.

The reasoning for converting a subnet using an inhibitor arc into one using debit arcs is as follows. The place s in net A is represented conceptually by a pair of places in net B, s and $\bar{s}$. Each time a token enters s, we record that fact by placing a debit in $\bar{s}$. To accomplish this, for each $t \in \bullet s$ we add a debit arc from $\bar{s}$ to t. We represent this simply by transition t_1 without loss of generality. Note that in translating a single inhibitor arc, we require a number of debit arcs equal to $|\bullet s|$.

We also want to record every time s loses a token. Let $s\circ$ be the set of transitions to which s sends an inhibitor arc. We record this by adding an arc from every $t \in s\bullet - s\circ$ to $\bar{s}$. For every $t \in s\circ$ we create arcs $\bar{s} \rightarrow t$ and $t \rightarrow \bar{s}$ as a locking mechanism. (This replicates the inhibiting fu· .ion of the inhibitor arc.)

We make the following stipulation for the initial marking. Place s should have the same token count in subnet B as it has in A. Place $\bar{s}$ should be marked so that

$$M(s) + M(\bar{s}) = 1 \qquad (INV)$$

where $M(s)$ will be a negative integer for a place s containing antitokens. Without loss of generality, in Figure 4 initially $M(s) = 0$, so initially $M(\bar{s}) = 1$.

We now show that nets A and B have equivalent behaviors. Since their transition sets are isomorphic, it is sufficient to show that the same preconditions and postconditions hold for the firing of each transition.

case 1: firing t_1
The preconditions for firing t_1 are the same with $\bullet t_1^{\mathrm{A}} = \bullet t_1^{\mathrm{B}} - \{\bar{s}\}$. It is the case that t_1^{B} has one more incident arc, the debit arc from $\bar{s}$. However, by definition this represents an always true condition; it can never prevent t_1 from firing. The postconditions for each with respect to s are the same: $M_f(s) = M_i(s) + 1$. Of course $M_f(\bar{s}) = M_i(\bar{s}) - 1$ even when $M_i(\bar{s}) \leq 0$ by definition.

We now show that INV holds:

$$M_i(s) + M_i(\bar{s}) = 1 \;\wedge\; M_i(s) = M_f(s) - 1 \;\wedge\; M_i(\bar{s}) = M_f(\bar{s}) + 1 \implies$$
$$(M_f(s) + 1) + (M_f(\bar{s}) - 1) = 1 \implies M_f(s) + M_f(\bar{s}) = 1$$

case 2: firing t_2
The preconditions are identical, $\bullet t_2^{\mathrm{A}} = \bullet t_2^{\mathrm{B}}$, so $M_i(s) > 0$ must hold. The postconditions are the same, $M_f(s) = M_i(s) - 1$, with, additionally, $M_f(\bar{s}) = M_i(\bar{s}) + 1$ in subnet B.

To show that INV holds:

$$M_i(s) + M_i(\bar{s}) = 1 \;\wedge\; M_i(s) = M_f(s) + 1 \;\wedge\; M_i(\bar{s}) = M_f(\bar{s}) - 1 \implies$$
$$(M_f(s) - 1) + (M_f(\bar{s}) + 1) = 1 \implies M_f(s) + M_f(\bar{s}) = 1$$

case 3: firing t_3
Note that $\bullet t_3^{\mathrm{A}} - s = \bullet t_3^{\mathrm{B}} - \bar{s}$. In A the precondition from s for firing c is $M(s) = 0$ as dictated by the inhibitor arc. The precondition from $\bar{s}$ in B is $M(\bar{s}) > 0$. But note that the following holds:

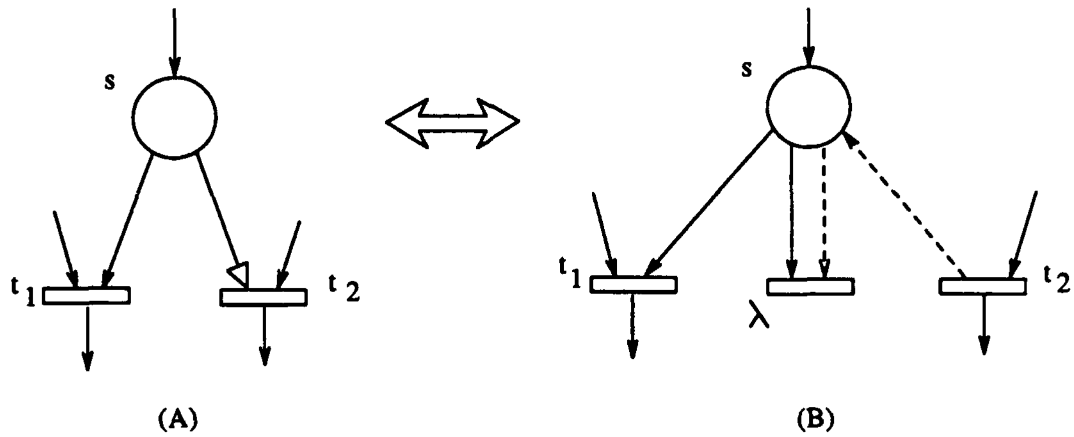


Figure 6: Representing debit arcs in two-color nets.

$$M(s) = 0 \wedge INV \implies M(\bar{s}) > 0 \ \ and$$
$$M(s) > 0 \wedge INV \implies M(\bar{s}) \leq 0$$

The preconditions are thus logically equivalent. (We need not consider the case of $M(s) < 0$ as this never occurs.)

To show that INV holds: note that in firing t_3 place s is not involved, so $M(s)$ cannot change. For the invariant to hold, $M(\bar{s})$ cannot change either. On firing t_3 a token is spent and a token is gained at $\bar{s}$, resulting in no net change.

The subnet A in Figure 4 universally represents the situational use of an inhibitor arc. We should mention there exists a degenerate case as seen in subnet A of Figure 5 where $s \in t_3\bullet \wedge t_3 \in s\circ$ (a self-loop containing an inhibitor arc). Our proof currently holds as we can restrict the discussion to only *pure* nets, which contain no *self-loops.* However, we also present a transformation for this special case in Figure 5. The two subnets here are equivalent. This rewrite was established by the same reasoning used to derive Figure 4 and can be proven by a proof virtually identical to above. □

3 Relation of debit nets to colored nets

To discuss debit nets as a case of colored nets, we revert to the more general pair notation to represent a debit net state. Likewise, we can consider the state of a 2-color net as a vector of pairs (c_1, c_2) where c_1 is the number of tokens of color 1 at a place and c_2 is the number of tokens of color 2. Conceptually, we can think of color 1 as being equivalent to tokens in the debit net and color 2 as antitokens.

Debit nets under delayed annihilation are equivalent to the standard class of place/transition nets. Token colors in nets have as well been shown to be a notational convenience that do not increase net power [2]. We can view debit nets as a special notation for a class of 2-color nets. This is easily seen by *folding* together places s and $\bar{s}$ from the normal net in Figure 2. We use two colors to distinguish the tokens of s and $\bar{s}$, representing tokens and antitokens respectively.

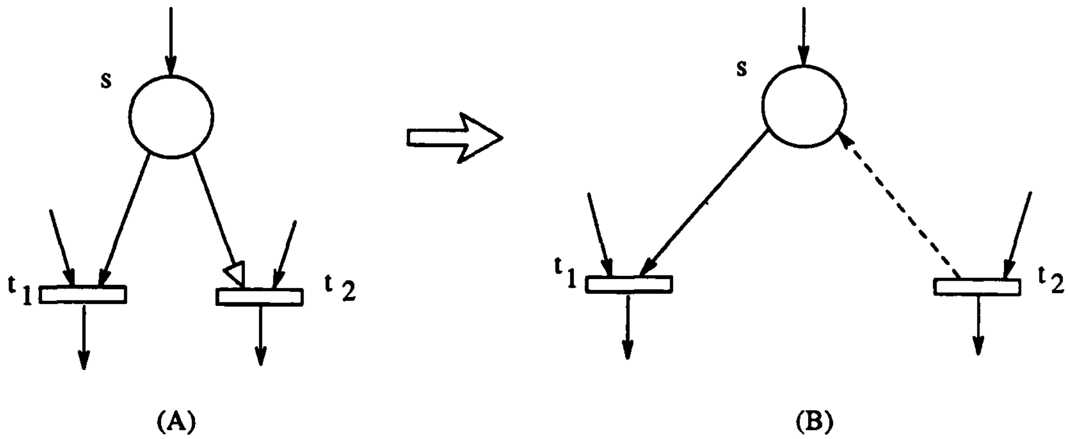


Figure 7: Debit arcs in two-color nets with instantaneous annihilation.

The equivalence is shown by Figure 6. In the colored net B the transition λ still simulates delayed annihilation. The same reasoning used earlier shows these subnets to have the same behavior.

Conversely, we can take the fact that the class of debit nets under the instantaneous annihilation rule is TM-equivalent to prove the following interesting conjecture about colored nets and annihilation:

Theorem 3:
The class of colored Petri nets using an instantaneous annihilation rule between any two given colors is TM-equivalent.

Proof:
We show this by providing a universal transformation from Petri nets with debit arcs under instantaneous annihilation (which are TM-equivalent) into a 2-color Petri nets employing an instantaneous annihilation rule between its two colors.

The transformation is represented in Figure 7. Using the formalism for instantaneous annihilation introduced earlier in Section 2.2, we now show the behavior of these two net constructs to be identical. Let $M(s)$ be the token count for color 1, and let $D(s)$ be the token count for color 2; let the 2-color net in the translation have an immediate annihilation policy between tokens of color 1 and color 2. The debit net immediately annihilates tokens and antitokens as defined. Set the initial marking for s in each subnet to the same pair, say (n, m). By the instantaneous annihilation policy it is immediately adjusted so that $(M(s), D(s))$ becomes $(n \dot{-} m, m \dot{-} n)$.

For this proof, we again have the convenience that the transition sets of A and B are isomorphic. Thus it suffices to show that the same preconditions and postconditions hold for the firing of each transition.

case 1: firing t_1
The preconditions in net each are the same. With respect to s we have that $M_i(s) > 0$ must hold. The postcondition for s in each case is

$$(M_f(s), D_f(s)) = (M_i(s) - 1, D_i(s)).$$

After annihilation adjustment they will be the same.

case 2: firing t_2
The preconditions are equivalent with $\bullet t_2^A - \triangleright t_2^A = \bullet t_2^B$. We have $\triangleright t_2^A = \{s\}$, a debit arc, which represents an always true condition. The postcondition for s in each subnet is

$$M_f(s) = (M_i(s), D_i(s) + 1).$$

In subnet A this is by definition of the debit arc, and in subnet B it is due to the arc $t_2 \rightarrow s$ producing a token of color 2.

Subnet A in Figure 7 universally represents the situational use of a debit arc.[4] By giving a universal reduction of debit nets under an instantaneous annihilation policy to colored nets with an annihilation policy we have shown the class of such colored nets to be TM-equivalent.
□

4 Observations and conjectures

Without formal proofs, we offer these observations and conjectures about the behavior of various types of debit nets.

First, one can imagine debit nets in which debts do not necessarily have to stay in the places at which they are made. Instead, antitoken movement rules can be created to transfer debt around the net. Useful antitoken movement rules can be defined for either annihilation policy we have discussed. One simple rule says that, in addition to other valid firings, a transition t that has one or more *antitokens* in each place in $\bullet t$ may fire and create an *antitoken* in each place in $t\bullet$. We conjecture that such a debt transfer rule does not change the recognition power of debit nets under either of the annihilation rules we have discussed.

Secondly, there are several classes of language that a debit net can be viewed as generating, depending not only on the normal properties of how symbols are mapped to transitions and how final states are defined, but on when transition symbols are inserted into the string being generated by debit net execution. These views exist regardless of the annihilation policy being used for execution:

1. A transition symbol can be added to the generated string when the transition fires, whether it creates a debt or not. This is the view analyzed in the earlier sections of this paper.

2. If firing a transition creates a debt, then its symbol is not added to the generated string until the debt is paid. Implementing this approach requires that the identity of the transition that makes the debt is retained with the antitoken.

3. If firing a transition creates a debt, then a symbol is not added to the generated string until the debt is paid. When a debt is paid, one of the transitions connected by debit arcs to the place at which *annihilation occurs* is arbitrarily chosen and added to the string.

[4]Note that unlike the earlier instant annihilation theorem, we do not need here to consider a self-loop involving a debit arc, $s \in t\bullet \wedge t \in s\triangleright$. Such a structure is semantically void and can simply be removed.

4. If firing a transition creates a debt, then a symbol is not added to the generated string until the debt is paid. When a debt is paid, one of the transitions connected by debit arcs to the place *at which the debt was made* is arbitrarily chosen and added to the string.

Note that the last two methods are actually the same for nets in which no antitoken movement is possible. This is because when there is no antitoken movement, the place at which the debt is made is the same as the one where the debt is paid.

References

[1] T. Agerwala. A complete model for representing the coordination of asynchronous processes. Computer Research Report 32, Johns Hopkins University, Computer Science Program, July 1974.

[2] K. Jensen. Coloured Petri nets and the invariant method. *Theoretical Computer Science*, 14:317–336, 1981.

[3] James L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Inc., 1981.

[4] Wolfgang Reisig. *Petri Nets: An Introduction*. Springer-Verlag, 1985.

UMIACS Technical Report Series

UMIACS–TR–90–33	**Double Coset Networks as Connectors for Parallel Processors** A. Yavuz Oruç
UMIACS–TR–90–34	**Permuting with Cosets of Symmetric Groups** Melda Y. Oruç and Yavuz A. Oruç
UMIACS–TR–90–35	**A Distributed User Information System** Steven D. Miller, Scott Carson, and Leo Mark
UMIACS–TR–90–36	**Yet Another Polynomial Preconditioner for the Conjugate Gradient Algorithm** Dianne P. O'Leary
UMIACS–TR–90–37	**Efficient Parallel Algorithms on the Network Model** Kwan Woo Ryu
UMIACS–TR–90–38	**A TCP Instrumentation and Its Use in Evaluating Roundtrip-Time Estimators** Dheeraj Sanghi, M.C.V. Subramaniam, A. Udaya Shankar, Olafur Gudmundsson, and Pankaj Jalote
UMIACS–TR–90–39	**Generalized Disjunctive Well-Founded Semantics for Logic Programs** Chitta Baral, Jorge Lobo, and Jack Minker
UMIACS–TR–90–40	**Recursive Star-Tree Parallel Data-Structure** Omer Berkman and Uzi Vishkin
UMIACS–TR–90–41	**Traffic Characterization of the NSFNET National Backbone** Steven A. Heimlich
UMIACS–TR–90–42	**Variable-Rate Finite-State Vector Quantization** Y. Hussain and N. Farvardin
UMIACS–TR–90–43	**A Structured Fixed-Rate Vector Quantizer from Variable-Length Encoded Scalar Quantizers** R. Laroia and N. Farvardin
UMIACS–TR–90–44	**Fast Modular Arithmetic on a Permutation Network** A. Yavuz Oruç, Vinod G.J. Peris, and M. Yaman Oruç
UMIACS–TR–90–45	**Permutation Networks Using Radix Sorting** Chingyuh Jan and A. Yavuz Oruç
UMIACS–TR–90–46	**Performance Study of Two Protocols for Voice/Data Integration on Ring Networks** Q. Yang, D. Ghosal, and S.K. Tripathi
UMIACS–TR–90–47	**Towards a Comprehensive Framework for Reuse: Model-Based Reuse Characterization Schemes** Victor R. Basili and H. Dieter Rombach